# Afleveringsopgave 2

Af Jesper Bertelsen: AU-ID: au689481

Et signal af formen

, ,

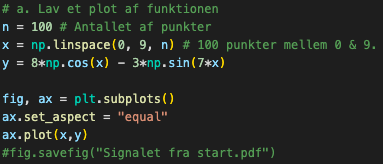
sendes igennem et kabel og forstyrres dermed af støj. Vi ønsker at modificere signalet, som er modtaget, så støjeffekten mindskes.

1. Lav en python plot af funktionen *𝑦*(*𝑥*) via f.eks. n = 100 *𝑥*-værdier jævnt fordelt over intervallet [0, 9].

Der laves en simpel x samling fra 0 ... 9, med 100 punkter.

Funktion af x laves som y og sammen plottes de.

Python kode:



Plottet:



Figure 1: Signalet fra start

1. Brug:

til at tilføje støj til 𝑦i formen.

Og plot resultatet

Hvilket jeg så har gjort.

Python kode:

Et billede, der indeholder tekst

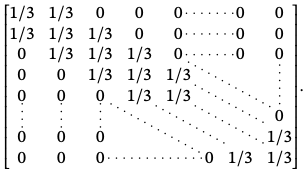
Automatisk genereret beskrivelse

Plottet:

**

Figure 2: Signalet med og uden støj

1. Givet en vektor v og et heltal offset vil funktionen np.diag(v, offset) danne en matrix med v langs en skrå linje hvor startpunktet er forskudt fra det øverste venstre hjørne med offset. Brug np.diag, eventuelt kombineret med np.ones(), tre gange til at konstruere matricen *𝐴* ∈ R*𝑛*×*𝑛*, der har formen



Python code:

Et billede, der indeholder tekst

Automatisk genereret beskrivelse

3 diagonaler skulle laves. 2 med n - 1 elementer, som er siderne. Midten har n antal elementer.

En variabel vægtning ganges på de 1’ere som er i vektoren, så der fås 1/3 i hver .

En matrice, A, laves så af summen af de diagonale matricer som laves til midten og siderne. Offset bestemmer søjle start, så til den første diag:

Søjle, j = -1, anden diag: j = 0, tredje: j = 1

1. Plot *𝐴𝑦*støj. Det burde ses at den har en form, som er tættere på *𝑦* end på *𝑦*støj. Forklar hvorfor.

Plot:



Figure 3: A matricen prikket på y

I mange af tilfældene er dette plot tættere på de den oprindelige værdi.

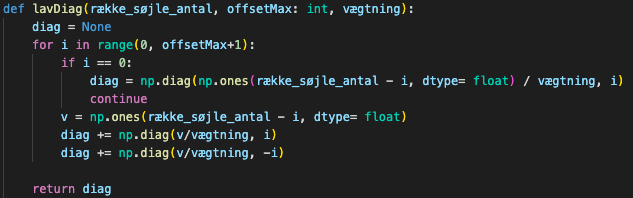
Hvorfor det er, ved jeg ikke.

1. Ved at ændre den relative vægtning af diagonalen i *𝐴* i forhold til de andre indgange, eller ved at bruge flere diagonaler som er ikke nul, giv et eksempel på en matrix *𝐵* hvor *𝐵𝑦*støj er har bedre støjreduktion end *𝐴𝑦*støj. Bekræft at dette gælder når der startes med en ny version of *𝑦*støj.

Måden som spørgsmålet forstås på er, at ved enten at ændre på værdierne for de diagonaler som haves eller ved at ændre i antallet af diagonaler, skal der findes en matrice, B, som har bedre støjreduction på y\_støj end A matricen fra tidligere opgaver, havde.

Den generelle procedure er gennemskuet, der laves en funktion til denne proces, så der nemt kan laves mange forskellige versioner af B.

Python code:



Der er ikke meget nyt i koden. For hver ændring i antal offset, skal vi lægge en diagonal til matricen, diag, med den offset værdi. Vægtningen varierer vi med inputtet vægtning.

For at fjerne noget af gentagelsen, så laver jeg en plotDiag funktion. Hvis eneste opgave er at plotte for x, matricen givet prikket med y.



Man kunne også bare lave diagonalen og plottet den i et, men det er muligvis godt nok også at have matricen. Det var det jeg gik med, så sådan blev det.

Konklusion:

======================================================================

*Efter at have gået systematisk til værks, kan der ikke findes en B, som resulterer i en bedre støjreduktion end A.*

*Hvad der blev prøvet var:*

* *Fastholde antal diagonaler, ændre vægtningen*
* *Fastholde vægtningen, ændre antal diagonaler.*
* *Behandle vægtningen og antal diagonaler som hinandens inverser, så x = antal diagonaler & 1/x = vægtningen. X = 1 er identitetsmatricen, som ikke giver støjreduktion, x = 5 er den næste efter x = 3 som A matricen havde. X = 5 medførte ikke til noget bedre resultat.*

*Hvis matricen skal være symmetrisk, som jeg har haft antaget, så har A den bedste støjreduktion.*

======================================================================

*Plot:*

**